

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-102289

(43)Date of publication of application : 13.04.1999

(51)Int.Cl.

G06F 9/06
G06F 9/445
G06F 13/00

(21)Application number : 09-262140

(71)Applicant : HITACHI SOFTWARE ENG CO LTD

(22)Date of filing : 26.09.1997

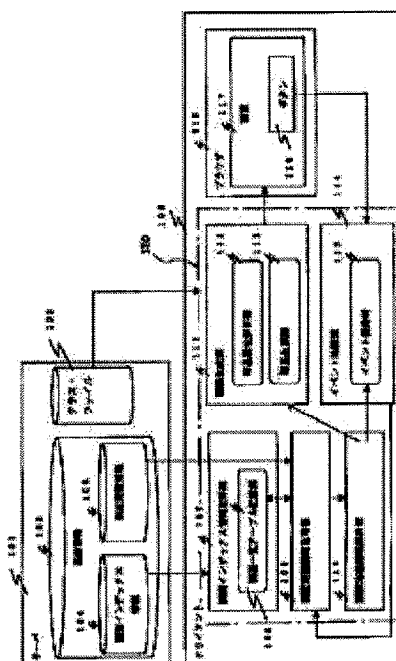
(72)Inventor : HOSOMI TERUHIRO
SHOMURA TSUTOMU

(54) SCREEN GENERATING METHOD FOR OPERATION PROCESS

(57)Abstract:

PROBLEM TO BE SOLVED: To shorten the wait time up to the start of operation processing irrelevantly the complexity of the operation processing and the number of screens by analyzing the screen defined in screen definition information by an analyzing function, generating components in the screen according to the analytic result by a screen generating function, and displaying them on the display screen.

SOLUTION: An applet registered in a class file 102 is equipped with an acquiring function for screen definition information, the analyzing function for the obtained screen definition information, the screen generating function based upon the analytic result, and an acquiring function for screen index information. This applet is downloaded previously to a client device 106 by a browser 116 before the operation processing and used to download screen definition information needed for the operation processing, screen by screen, in order as the screen transition advances, thereby generating the display contents of the screens according to the downloaded screen definition information.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-102289

(43) 公開日 平成11年(1999) 4月13日

(51) Int.Cl. ⁶	識別記号	F I
G 0 6 F 9/06	5 3 0	G 0 6 F 9/06 5 3 0 P
9/445		13/00 3 5 7 Z
13/00	3 5 7	9/06 4 2 0 J

審査請求 未請求 請求項の数5 O L (全 14 頁)

(21) 出願番号 特願平9-262140

(22) 出願日 平成9年(1997) 9月26日

(71) 出願人 000233055

日立ソフトウェアエンジニアリング株式会
社

神奈川県横浜市中区尾上町6丁目81番地

(72) 発明者 細美 彰宏

神奈川県横浜市中区尾上町6丁目81番地
日立ソフトウェアエンジニアリング株式会
社内

(72) 発明者 正村 勉

神奈川県横浜市中区尾上町6丁目81番地
日立ソフトウェアエンジニアリング株式会
社内

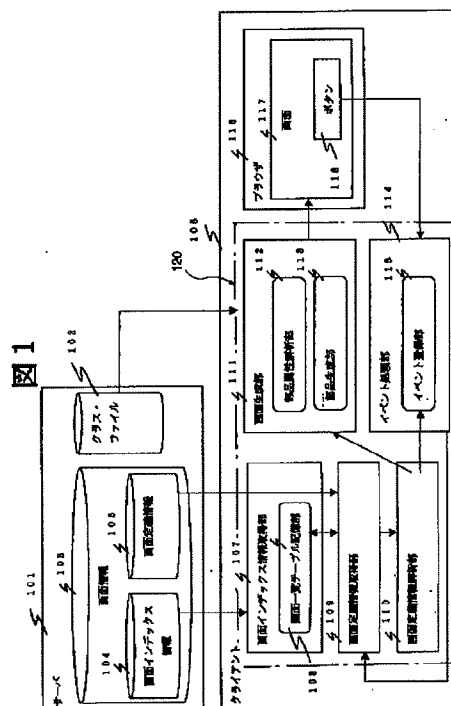
(74) 代理人 弁理士 秋田 収喜

(54) 【発明の名称】 業務処理用画面生成方法

(57) 【要約】

【課題】 業務処理の複雑さや画面数に関係なく業務処理開始までの待ち時間を短くし、各種の業務処理を効率良く進められるようにすること。

【解決手段】 業務処理を開始するに際し、サーバ装置側に用意された全ての画面や処理をクライアント装置側に一挙にダウンロードするのではなく、サーバ装置側に用意されている画面のそれぞれを定義した画面定義情報と、画面定義情報に従って画面を生成する生成機能および画面定義情報の取得機能、解析機能を有するアプレットのみを最初にダウンロードし、業務処理の進行に従って必要になる画面等の情報はその都度ダウンロードする。



【特許請求の範囲】

【請求項1】 所定のプログラミング言語で記述されたアプレットをサーバ装置からクライアント装置にダウンロードし、そのアプレットを用いて各種の業務処理で使用する画面を生成する方法であって、サーバ装置側に、業務処理で使用する画面の定義情報を設けると共に、この画面定義情報の取得機能および画面定義情報の解析機能と画面生成機能を有するアプレットのクラスファイルを設けておき、クライアント装置における業務処理の開始に先立ち、前記画面定義情報の取得機能および画面定義情報の解析機能と画面生成機能を有するアプレットをサーバ装置からクライアント装置にダウンロードし、そのダウンロードしたアプレットの画面定義情報の取得機能により業務処理で使用する所望の画面の画面定義情報を画面単位でサーバ装置からダウンロードして取得し、その画面定義情報で定義されている画面を前記解析機能により解析し、その解析結果に応じて画面内の部品を前記画面生成機能により生成し、クライアント装置の表示画面に表示させることを特徴とする業務処理用画面生成方法。

【請求項2】 複数の業務処理に関する画面定義情報のインデックス情報をサーバ装置に格納し、所望の業務処理名に対応するインデックス情報をサーバ装置からクライアント装置にダウンロードし、そのインデックス情報で定義されている画面定義情報を画面単位でダウンロードすることを特徴とする請求項1記載の業務処理用画面生成方法。

【請求項3】 前記画面定義情報は、表示する画面上の部品およびその属性、画面上でのイベント処理の内容を含むものであることを特徴とする請求項1または2記載の業務処理用画面生成方法。

【請求項4】 生成した画面上でイベントが発生した際に、そのイベントが画面遷移かを判定し、次画面に遷移するイベントであれば、次画面に該当する画面定義情報を検索し、遷移するイベントでなければ、該当するイベントを実行することを特徴とする請求項3記載の業務処理用画面生成方法。

【請求項5】 ダウンロードした画面インデックス情報を画面一覧テーブルに記憶し、そのインデックス情報で示される複数の画面について既に生成した画面か否かをフラグによって区別し、既に生成済みの画面については記憶された生成済みの画面を表示することを特徴とする請求項2記載の業務処理用画面生成方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、JavaアプレットなどのWWWブラウザ上で動作するプログラムを用いて給与計算などの各種の業務処理に使用する画面を生成する業務処理用画面生成方法に関するものである。

【0002】

【従来の技術】アプレットとは、WWW (World Wide Web) ブラウザ上で動作する小さなプログラムのことであり、クライアント装置に実装されているWWWブラウザがインターネットを介してWWWサーバから読み込み、クライアント装置上で実行する。このアプレットは、バイトコードのクラスファイルから成っており、そのプログラミングのためにJavaというプログラミング言語が提供されている。

【0003】Java言語は、オブジェクト思考言語であり、マルチプラットフォームで動作可能であるなどの特徴を持っている。この特徴を活用すれば、アーキテクチャが異なるクライアント装置であっても、例えば給与計算を行なうアプレットをWWWサーバに用意しておき、それぞれのクライアント装置で給与計算処理が必要になった時に、給与計算のアプレットをWWWサーバから読み込んで（ダウンロードして）実行することにより、全く同じ仕様の計算結果を得ることができるよう、アーキテクチャが異なることをユーザに意識させることなく必要な操作を行なわせることができる。そのため、アーキテクチャが異なる各種のクライアント装置を使用している企業等において業務処理への応用が盛んになっている。

【0004】インターネットやイントラネットにおいて業務処理をアプレットで実現する際には、業務処理で必要となる画面や処理内容を記述したアプレットのクラスファイルをWWWサーバにおいておき、このクラスファイルを業務処理で必要になった時に、クライアント側にダウンロードして実行するように構成するのが一般的である。

【0005】図10は、サーバ装置1001とクライアント装置1003とからなるシステム構成において、Javaアプレットの実行動作を説明する図であり、サーバ装置1001には、アプレットのクラスファイル1002が予め作成されており、クライアント装置1003側でブラウザ1005を実行し、アプレットが貼り付けてあるページを開いたときに、アプレットのクラスファイル1002がネットワーク経由でサーバ装置1001側からクライアント装置1003側にダウンロードされ、クライアント装置1003側の画面・処理1004が画面の生成などの処理を行ない、ブラウザ1005の画面1006を表示する等、当該アプレットで記述された処理を進める。

【0006】アプレットで必要となる画面や処理は、クラスファイル1002として作成されており、アプレットの実行時にこのクラスファイルを解釈して画面の表示や処理の実行が行われる。

【0007】ところが、アプレットを企業等で使用する各種の業務処理に適用した場合、業務処理で使用する画面の数が多く、また各画面内での処理内容が複雑になる場合が多い。画面数が増加し、また処理内容が複雑にな

れば、画面数および処理内容の複雑さに比例してアプレットの規模が大きくなり、サーバ装置 1001 からのダウンロード時間が長くなり、業務処理を実際に開始するまで長時間待たされるという問題がある。

【0008】そこで、アプレットのクラスファイル 1002 を圧縮してダウンロードすることにより、ダウンロード時間を短縮する方法を実施している例がある。

【0009】

【発明が解決しようとする課題】しかし、業務処理に適用した場合、アプレットの規模はかなり大きいため、圧縮したとしても、ダウンロード時間は僅かしか短縮されず、実際に業務処理を開始するまでに、業務処理内容の複雑さや画面数に比例した（アプレットの規模に比例した）時間だけ待たされるという問題は基本的には解決されていない。

【0010】一方、アプレットを作成するためのツールなど、数多く開発されているが、業務処理中のある画面に対してボタンなどの部品を追加、削除などの修正を行なった場合、その都度、アプレットを再作成し、テストするといった作業が発生し、業務処理の変更に対して柔軟に対処できないという問題がある。

【0011】本発明の第 1 の目的は、業務処理の複雑さや画面数に関係なく業務処理開始までの待ち時間を短くすることができる業務処理用画面生成方法を提供することにある。

【0012】また、第 2 の目的は、業務処理の画面や処理内容の変更に対して柔軟に対処することができる業務処理用画面生成方法を提供することにある。

【0013】

【課題を解決するための手段】上記第 1 の目的を達成するために本発明は、サーバ装置側に、業務処理で使用する画面の定義情報を設けると共に、この画面定義情報の取得機能および画面定義情報の解析機能と画面生成機能を有するアプレットのクラスファイルを設けておき、業務処理の開始に先立ち、前記画面定義情報の取得機能および画面定義情報の解析機能と画面生成機能を有するアプレットをサーバ装置からクライアント装置にダウンロードし、そのダウンロードしたアプレットの画面定義情報の取得機能により業務処理で使用する所望の画面の画面定義情報を画面単位でサーバ装置からダウンロードして取得し、その画面定義情報で定義されている画面を前記解析機能により解析し、その解析結果に応じて画面内の部品を前記画面生成機能により生成し、クライアント装置の表示画面に表示させるようにしたことを特徴とする。

【0014】また、第 2 の目的を達成するために、サーバ装置側に、複数の業務処理のそれぞれで使用する画面の一覧をインデックス情報として用意しておき、業務処理の開始に先立ち、所望の業務処理名のインデックス情報をサーバ装置からクライアント装置にダウンロード

し、そのダウンロードしたインデックス情報で定義されている画面の画面定義情報を画面単位でサーバ装置からダウンロードして取得し、その画面定義情報で定義されている画面を前記解析機能により解析し、その解析結果に応じて画面内の部品を前記画面生成機能により生成し、クライアント装置の表示画面に表示させるようにしたことを特徴とする。

【0015】すなわち、本発明は、業務処理を開始するに際し、サーバ装置側に用意された全ての画面や処理をクライアント装置側に一挙にダウンロードするのではなく、サーバ装置側に用意されている画面のそれぞれを定義した画面定義情報と、画面定義情報に従って画面を生成する生成機能および画面定義情報の取得機能、解析機能を有するアプレットのみを最初にダウンロードし、業務処理の進行に従って必要になる画面等の情報はその都度ダウンロードするようにし、業務処理開始までの待ち時間を画面定義情報と画面生成機能、画面定義情報の取得機能および解析機能のダウンロード時間のみに限定するようにしたことを特徴とする。

【0016】

【発明の実施の形態】以下、本発明の一実施形態について図面を用いて説明する。図 1 は、本発明を適用した業務処理画面生成システムの実施形態を示すシステム構成図である。この実施形態のシステムは、基本的には、サーバ装置 101 とクライアント装置 106 とから成るクライアント・サーバシステムである。

【0017】図 1 において、102 はアプレットを成すクラスファイルであり、クライアント装置 106 のブラウザからアプレットが要求された際にサーバ装置側 101 からクライアント 106 側にネットワーク経由でダウンロードされる。クラスファイル 102 に登録されているアプレットは、業務処理毎に用意される画面インデックス情報の取得機能、業務処理で使用する画面定義情報の取得機能、取得した画面定義情報の解析機能、解析結果に基づく画面生成機能を備えるものである。画面インデックス情報の取得機能は業務処理が一つの場合には、画面インデックス情報が必要ないので無くてもよい。本実施形態におけるクラスファイル 102 に登録されたアプレットは、上記の 4 つの機能、すなわち画面定義情報の取得機能、取得した画面定義情報の解析機能、解析結果に基づく画面生成機能、画面インデックス情報の取得機能を備えている。

【0018】103 は、複数の業務処理で使用する各種の画面の一覧を定義している画面情報であり、画面インデックス情報 104 と画面定義情報 105 とから成っている。画面定義情報 105 は、アプレットを用いて表示する画面の内容を定義した情報であり、1 画面ごとに格納されている。画面インデックス情報 104 は、画面定義情報 105 の一覧を記述した情報である。

【0019】一方、クライアント装置 106 には、サー

バ装置101から各種の業務処理に必要なアプレットをダウンロードするためのブラウザ116が設けられており、このブラウザ116によってクラスファイル102に登録されたアプレットがネットワーク経由でダウンロードされる。クラスファイル102に登録されたアプレットは、業務処理を行なう前に、ブラウザ116によってクライアント装置106に予めダウンロードされ、そのダウンロードしたアプレットを用いて、業務処理に必要な画面定義情報を画面単位で、画面遷移の進行に合わせて順次ダウンロードし、そのダウンロードした画面定義情報に基づき画面の表示内容を生成するようになって

【0020】従って、ブラウザ116によって上記の機能を備えたアプレット120がダウンロードされたことにより、クライアント装置106は、図1に示すように、画面インデックス情報取得部107、画面定義情報取得部109、画面定義情報解析部110、画面生成部111、イベント処理部114という機能を有することになる。

【0021】以下、これらの各部の構成および動作について説明する。107は、画面インデックス情報104をネットワーク経由で取得する手段である画面インデックス情報取得部である。108は、画面インデックス情報104および、それぞれの画面の状態を管理するための画面一覧テーブル記憶部である。画面インデックス情報取得部107は画面インデックス情報104をサーバ装置101側からネットワーク経由で取得し、画面一覧テーブル記憶部108に設定する。

【0022】109は、画面定義情報105をネットワーク経由で取得する手段である画面定義情報取得部であり、画面一覧テーブル記憶部108から生成表示対象の画面定義情報105を検索し、サーバ装置101側からネットワーク経由で画面定義情報105を取得する。110は画面定義取得部109で取得した画面定義情報105の内容を解析する画面定義情報解析部であり、解析して得られた情報が部品情報かイベント情報かを判定し、該当する処理を行う。

【0023】111は画面定義情報解析部110で得られる情報が部品情報の場合に、部品生成の処理をする画面生成部であり、この画面生成部111は部品属性解析部112および部品生成部113で構成される。画面定義情報解析部110で得られた情報が部品情報であれば、画面生成部111が呼ばれ、部品属性解析部112が部品情報の属性部分を解析し、得られた情報の内容をもとに部品生成部113が部品を生成する。

【0024】114は画面定義情報解析部110で得られる情報がイベント情報の場合に、イベント登録の処理をするイベント処理部である。また、イベント処理部114は画面上でイベントが発生した際のイベントの判定および処理を行う。

【0025】115は画面上のイベントを格納しておくイベント登録部である。画面定義情報解析部110で得られた情報がイベント情報であれば、イベント処理部114が呼ばれ、イベント登録部115に画面上のイベントとして設定する。

【0026】画面定義情報105のすべての部品情報およびイベント情報に対して画面定義情報解析部110での処理が終了すると、ブラウザ116で実行されるアプレット上に画面117として表示する。

【0027】118は画面117上に生成されたボタンである。ボタン118が押された際にイベントが発生し、イベント処理部114により、その発生したイベントの判定および処理を行い、イベント登録部115から該当するイベントの処理を検索し実行する。また、イベントが画面遷移である場合、次画面を生成し表示する処理を行う。これは、画面定義情報取得部109を呼び出し、次画面として指定された画面定義情報105を取得し、以下同じ処理を行い、次画面の生成の処理を行う。

【0028】図2は、1つの画面に表示する部品等の情報を定義した画面定義情報105の構成内容を示す図である。図2において、201は表示する部品の種類を示すキーを格納するフィールドであり、部品キー201にはラベルやテキスト、ボタンなどを指定する。画面生成部111に画面を構成する必要な部品に対応する部品キー201を登録しておく。

【0029】202は画面上で行う処理で部品を特定するための部品IDである。203から205は部品キー201に対する部品の属性情報を指定する部分であり、属性情報には表示文字列や表示位置、サイズ、状態、処理などがあり、複数の属性情報を連ねて指定するようになっている。以降、同じように画面上に表示する部品の個数分だけ定義情報を格納する。

【0030】206は画面上で処理される際に使用されるイベント名である。画面定義情報解析部110では部品キー201に該当しないものをイベント名206として識別する。207はイベント名206に対するイベント処理の内容であり、画面上でイベントが起きた場合、イベント処理部114ではイベント名206をキーワードとしてイベント内容207を検索し、イベント名206に対応するイベント内容207を処理する。

【0031】図3は、画面定義情報105の内容である部品の属性203、204、205の構成内容を示す図であり、301は表示する部品の属性の種類を示すキーを格納するフィールド、302は属性キー301に対する属性内容を指定する部分である。属性キー301には、表示文字列や部品の表示位置およびサイズや色、状態、処理するイベント名などを指定する。例えば、ボタン部品の属性である処理208を「画面遷移」にする場合、処理208に関する属性キー301＝処理、属性内容＝画面遷移という内容に設定される。

【0032】図4は、給与計算や勤務管理などの各種の業務処理において必要となる画面を画面遷移順にあるいは順不同で画面一覧として格納している画面インデックス情報104の構成内容を示す図である。図4において、401は1つの業務処理の画面インデックス情報内に格納する画面数を格納するフィールドである。402はそれぞれの画面の名称を指定するための画面名称を格納するフィールドである。例えば、給与計算を行なう業務処理において、業務開始から業務終了までに必要とする画面の数がm面であった場合、画面数401にはmが格納され、画面名称402にはm個の画面名称が格納される。403は画面名称402の一覧から業務処理の初期表示に使用する画面を指定する部分であり、画面名称402の中の1つが選択されて格納される。404は画面名称402と対応する画面定義情報の格納場所を指定する情報を格納するフィールドであり、具体的には、画面名称402と対応する画面定義情報が、サーバ装置101内において実際に格納されている場所を指す画面定義情報名を格納するフィールドである。画面名称402と画面定義情報名404の一覧の個数が画面数401に設定される。

【0033】このように構成される画面インデックス情報104は、1つの業務種別について1組存在する。すなわち、業務処理として、給与計算と勤務管理との2種類があった場合、画面インデックス情報104は2組設けられる。

【0034】図5は、画面インデックス情報104の内容を取得し、画面一覧テーブル記憶部108に作成される画面一覧テーブルの構成内容を示す図である。図5において、501は画面インデックス情報104の構成内容である画面名称402が格納されるフィールドである。502は画面名称402に対応する画面定義情報名404が格納されるフィールドである。503と504は画面の状態を管理するためのフィールドであり、503は必要な画面が既にアプレット120上に生成されたかどうかを示すフラグであり、初期設定では、まだ生成されていないことを示す“FALSE”が設定されている。504は画面上のテキストの文字列やチェックボックスの選択状態をその部品に付けられた部品ID202と対応させて格納するフィールドであり、初期設定では、まだ何も入力、選択されていないことを示す“NULL”が設定されている。

【0035】図6は、2つの業務処理に対する画面インデックス情報104と画面定義情報105の構成を示す図であり、ここでは2種類の業務処理に使用する画面インデックス情報と画面定義情報の関係を示している。図6において、601は給与計算の業務処理を定義した画面インデックス情報であり、602は勤務管理の業務処理を定義した画面インデックス情報である。これらのうちのどちらか一方を画面インデックス情報取得部107

によりクライアント装置106内に取得することにより、その取得した画面インデックス情報で定義している業務処理を実行することができる。603は幾つかの画面定義情報を定義した画面定義情報群であり、ここでは給与計算および勤務管理の業務処理を定義した画面インデックス情報601、602に記述されている画面定義情報をダウンロードしたものである。これらの画面定義情報は、2つの業務処理に共通の開始画面、例えばパスワード入力画面604を共用し、画面インデックス情報601、602のいずれかを指定することによりダウンロードすることができる。また、給与計算の業務処理の画面構成を変更する必要がある場合は、給与計算の画面インデックス情報601および画面定義情報を変更し、その変更後の画面インデックス情報601および画面定義情報をダウンロードすることにより、変更後の画面構成での給与計算業務を行なうことが可能になる。すなわち、従来にあっては、業務に使用する画面の構成を変更する必要がある場合、その業務に使用するアプレット自体に変更を加えていたが、本実施形態においては、画面定義情報のみを変更するだけでよく、画面構成や処理内容の変更に対して柔軟に対処できるのである。

【0036】図7は、画面インデックス情報取得部107の処理の流れを示すフローチャートである。以下、図7を用いて画面インデックス情報104をサーバ装置101側からネットワーク経由で取得して、画面一覧テーブル記憶部108に画面一覧テーブルを作成する動作を説明する。

【0037】画面インデックス情報取得部107は、アプレット（クラスファイル102から業務開始前にダウンロードされたアプレット120）が起動された時点で呼び出される。まず、アプレット上で実行する業務処理として指定された画面インデックス情報104の名称を取得する（ステップ701）。例えば、「給与計算」の業務処理に対応する画面インデックス情報104の名称を取得する。この画面インデックス情報104の名称は、アプレットを実行する際にパラメータとしてパスワード入力画面から入力して渡すなどの方法をとる。

【0038】この後、指定された名称の画面インデックス情報104の内容をサーバ装置101側からネットワークを介して取得する（ステップ702）。次に、取得した画面インデックス情報104から画面数401を取得し（ステップ703）、その取得した画面数401により、画面一覧テーブルのサイズを決め、画面一覧テーブル記憶部108に、そのテーブルの領域を確保する（ステップ704）。

【0039】次に、アプレットで最初に表示する初期画面403の画面名称402を画面インデックス情報104から取得する（ステップ705）。そして、初期画面403の画面名称403に対応する画面定義情報名404を画面インデックス情報104から取得し（ステップ

706)、図5の画面一覧テーブルに初期状態で設定する(ステップ707)。初期状態として、画面生成フラグ503には、まだ画面を生成していないことを示す"FALSE"を設定し、また入力データ504には画面上にデータはまだ入力、または選択されていないことを示す"NULL"を設定する。

【0040】画面インデックス情報取得部107は、画面一覧テーブルに未登録の画面名称402および画面定義情報名404がまだ残っているかどうかを調べる(ステップ708)。もし、ある場合にはステップ706からの処理を繰り返し、ない場合には処理を終了する。画面インデックス情報104の内容である画面数401は、画面名称402と画面定義情報名404と同じ個数か、多い個数を指定しておく必要がある。これにより、画面数401で指定された数の画面を構成する画面定義情報名502、画面生成フラグ503、入力データ504が図5の画面一覧テーブルに登録される。この場合、各画面名称の画面生成フラグ503および入力データ504には、初期状態であるので、それぞれFALSE、NULLが設定される。

【0041】図8は、画面定義情報取得部109および画面定義情報解析部110、画面生成部111、イベント処理部114の処理の流れを示すフローチャートである。

【0042】以下、図8を用いてアプレット120を用いてブラウザ116の画面117を生成する動作について説明する。画面定義情報取得部109は、画面定義情報105をサーバ装置101側からネットワークを介して取得する。画面定義情報解析部110は画面定義情報取得部109が取得する画面定義情報105の内容である部品情報またはイベント情報が1つ入力されるごとに呼び出される。また、生成する画面名称は、画面定義情報取得部109が画面遷移のイベントが発生する都度、イベント処理部114から呼ばれる際にイベント処理部114から渡される。

【0043】まず、画面定義情報取得部109は、画面一覧テーブル記憶部108の画面一覧テーブルから画面名称501に対応する画面定義情報名502を検索する(ステップ801)。そして、その検索して取得した画面定義情報名502で示される画面定義情報105をネットワーク経由でサーバ装置101からダウンロードして取得する(ステップ802)。次に、取得した画面定義情報105の内容から部品またはイベントの情報を取得する(ステップ803)。

【0044】画面定義情報解析部110は、画面定義情報取得部109が取得した情報の内容の先頭部分を取り出し、取得した情報が部品情報かイベント情報かを調べる(ステップ804)。もし、部品情報であれば、画面生成部111を呼び出す。画面生成部111は、部品情報から部品キー201を取り出し、その種類を判定す

る。そして、部品属性解析部112によって部品に対する属性203、204、205を切り出し、属性キー301とその属性内容302を解析する(ステップ805)。

【0045】部品生成部113は、ステップ805で解析した部品キー201に該当する部品を属性203、204、205に従って生成する(ステップ806)。

【0046】一方、画面定義情報取得部109が取得した情報がイベント情報であれば、そのイベント内容をイベント処理部114によりイベント登録部115に登録する(ステップ807)。

【0047】以上の部品生成またはイベント登録が終わると、画面定義情報解析部110は画面定義情報105の内容中に次の部品またはイベントの情報があるかどうかを調べる(ステップ808)。もし、ある場合にはステップ803からの処理ステップを繰り返す。ない場合には、画面上の全ての部品生成およびイベント登録が終了したものと判定する。

【0048】次に、画面一覧テーブルの該当する画面生成フラグ503に"TRUE"を設定し、この画面は生成済みであることを示すフラグを設定する(ステップ809)。そして、生成した画面をアプレットに画面名称を付加して追加する(ステップ810)。次回、既に生成した画面について、この画面処理を呼び出す前に、画面一覧テーブルの該当する画面生成フラグ503をチェックすることによって、同じ画面を繰り返し生成することを防ぐことができる。既に生成した画面であれば、画面名称によって該当する画面を表示する。

【0049】図9は、イベント処理部114の処理の流れを示すフローチャートである。以下、図9を用いて業務処理における画面を遷移させる動作を説明する。イベント処理部114は、発生したイベントを判定して該当する処理を行う。例えば、ブラウザ116の画面117上にボタン118があり、このボタンがクライアント装置106に付属したマウスまたはキーボードによって選択されたとする。すると、ボタン118が選択されたことを示すイベントが発生する(ステップ901)。イベント処理部114は、このイベントが何を行うかを判定する。例えば、ボタン118の部品の属性208としてボタンが処理するイベント名を指定しておく。イベント名208からイベント登録部115に登録してあるイベント名206を検索し、このイベント名206に対応するイベント内容207が、画面を遷移させるイベントかどうかを調べる(ステップ902)。もし、画面を遷移させるイベントでない場合、イベント登録部115に登録してある該当するイベントを処理し、終了する(ステップ903)。

【0050】例えば、イベント登録部115に登録してある処理が画面上のデータを取得する処理であれば、画面上の各部品に入力されているデータや選択状態を取得

し、各部品に付けられた部品ID202とともに、画面一覧テーブルの入力データ504に格納する。もし、画面を遷移させるイベントであれば、遷移先の画面名称501を画面一覧テーブル記憶部108の画面一覧テーブルから取得する(ステップ904)。画面一覧テーブルから該当する画面名称501を検索したならば、画面生成フラグ503を調べ、この画面が既に生成済みかどうかを確認する(ステップ905)。もし、生成していない場合は、画面名称501をパラメータとして画面生成部111を呼び出す(ステップ906)。既に生成済みであるならば、アプレット上に既に追加されている画面を画面名称によって判定し、表示する。画面を生成した場合は、それをそのまま表示する(ステップ907)。

【0051】このように既に生成済みの画面については、再利用する仕組みを設けておくことにより、画面遷移の都度、必要な画面の画面定義情報105をサーバ装置101からダウンロードする必要がなくなり、ネットワークのトラフィックが増大するのを防止することができる。

【0052】なお、本発明は、上記実施形態に限定されるものではなく、クライアント装置106に組み込まれた他のアプリケーションプログラムまたは他のアプレットと動的または静的に結合して実行させるようにしてもよい。

【0053】

【発明の効果】以上のように本発明によれば、業務処理を開始するに際し、サーバ装置側に用意された全ての画面や処理をクライアント装置側に一挙にダウンロードするのではなく、サーバ装置側に用意されている画面のそれぞれを定義した画面定義情報と、画面定義情報に従って画面を生成する生成機能および画面定義情報の取得機能、解析機能を有するアプレットのみを最初にダウンロードし、業務処理の進行に従って必要になる画面等の情報はその都度ダウンロードするようにしたため、業務処理開始までの待ち時間を画面定義情報と画面生成機能、画面定義情報の取得機能および解析機能のダウンロード時間のみに限定されることになり、業務処理の複雑さや画面数に関係なく業務処理開始までの待ち時間が短くなり、各種の業務処理を効率良く進めることが可能になる。

【0054】また、サーバ装置側に、複数の業務処理のそれぞれで使用する画面の一覧をインデックス情報として用意しておき、業務処理の開始に先立ち、所望の業務処理名のインデックス情報をサーバ装置からクライアン

ト装置にダウンロードし、そのダウンロードしたインデックス情報で定義されている画面の画面定義情報を画面単位でサーバ装置からダウンロードして取得し、その画面定義情報で定義されている画面を前記解析機能により解析し、その解析結果に応じて画面内の部品を前記画面生成機能により生成し、クライアント装置の表示画面に表示させることにより、業務処理に使う画面の変更に際しては画面インデックス情報および画面定義情報を変更するのみで対処することができ、業務処理の画面や処理内容の変更に對して柔軟に対処することができる。

【0055】また、画面定義情報はアプレットからの利用だけではなく、Java言語以外でプログラミングしたものからも利用可能であり、多言語からの移植が容易に行えるなどの効果がある。

【図面の簡単な説明】

【図1】本発明を適用した業務処理画面生成システムの実施形態を示すシステム構成図である。

【図2】画面定義情報の構成例を示す図である。

【図3】画面定義情報の内容である部品に対する属性の構成例を示す図である。

【図4】画面インデックス情報の構成例を示す図である。

【図5】画面一覧テーブル記憶部に作成される画面一覧テーブルの構成例を示す図である。

【図6】異なる業務処理での画面インデックス情報と画面定義情報の構成例を示す図である。

【図7】画面インデックス情報から画面一覧テーブルを作成する処理の流れを示すフローチャートである。

【図8】画面定義情報から部品情報を抽出して画面上の部品を生成する処理の流れを示すフローチャートである。

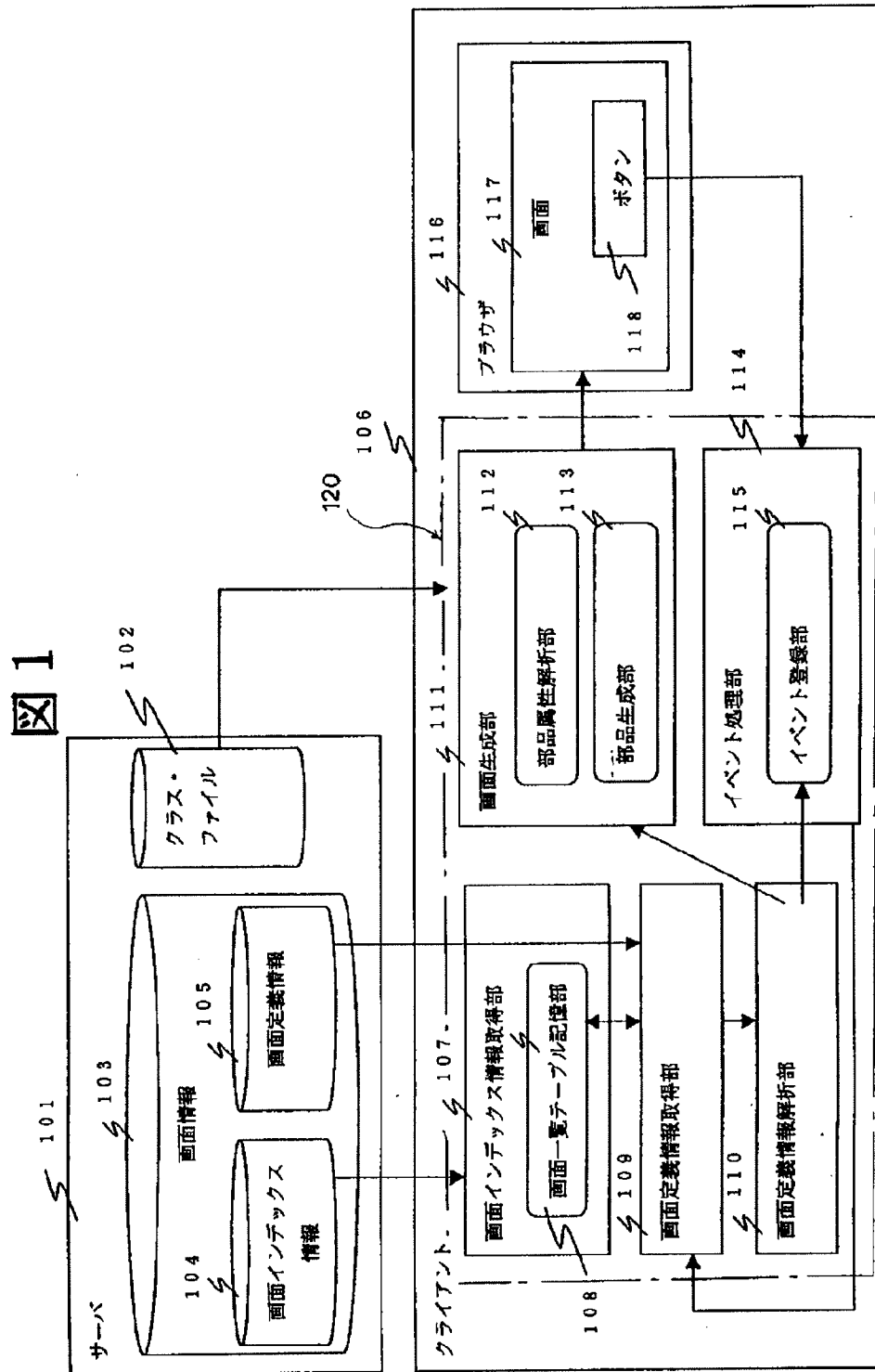
【図9】発生したイベントを判定して画面遷移ならば画面生成処理を行うイベント処理部の処理の流れを示すフローチャートである。

【図10】アプレットの実行動作を示す図である。

【符号の説明】

101…サーバ装置、102…クラスファイル、103…画面情報、104…画面インデックス情報、105…画面定義情報、106…クライアント装置、107…画面インデックス情報取得部、108…画面一覧テーブル記憶部、109…画面定義情報取得部、110…画面定義情報解析部、111…画面生成部、112…部品属性解析部、113…部品生成部、114…イベント処理部、115…イベント登録部、116…ブラウザ。

【図1】



【図2】

図2

105

201	202	203	204	205
部品キー	部品ID	属性	属性	属性
ラベル	L001	表示文字列	表示位置	..
テキスト	T001	表示位置	表示サイズ	状態
:				
206	207			
ボタン	B004	表示位置	表示色	処理
イベント名	イベント内容			
画面遷移	画面W0A002に遷移			
データ取得	現在の画面上のデータを取得			
:				

【図3】

図3

属性キー	属性内容
------	------

【図4】

図 4

104

401

画面数 : 15	
初期画面 : W0A002	
画面名称	画面定義情報名
W0A002	APP-WINDOW/R001/W0A002.DEF
:	
W0B015	APP-WINDOW/R002/W0B015.DEF

402

403

404

【図5】

図 5

108

501

502

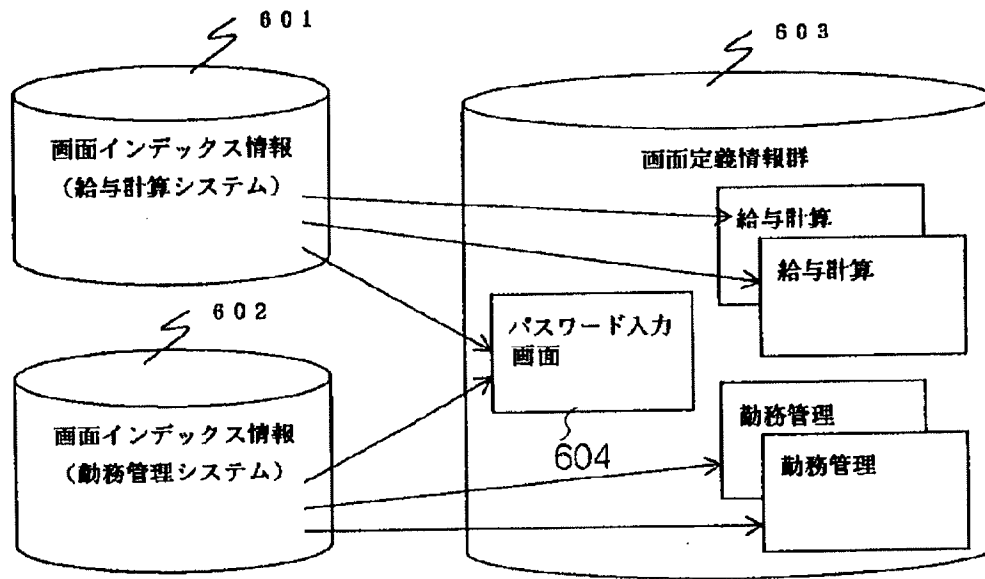
503

504

画面名称	画面定義情報名	画面生成フラグ	入力データ
W0A002	APP-WINDOW/R001/W0A002.DEF	FLASE	NULL
:			
W0B015	APP-WINDOW/R002/W0B015.DEF	FLASE	NULL

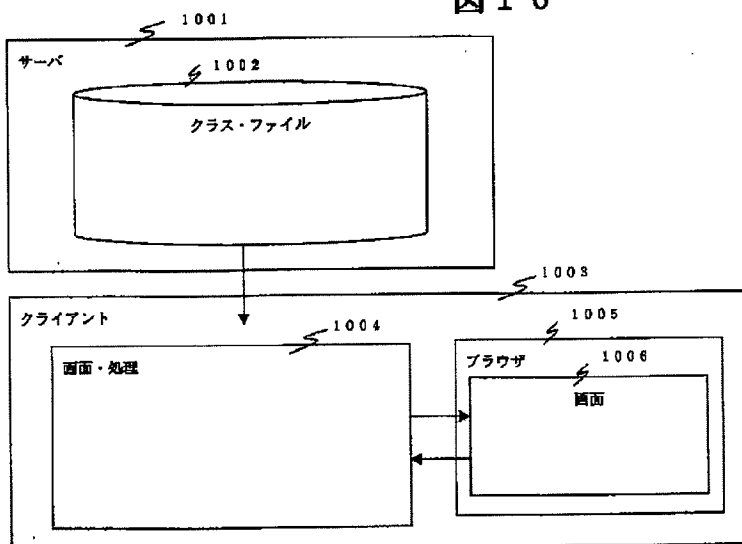
【図6】

図 6



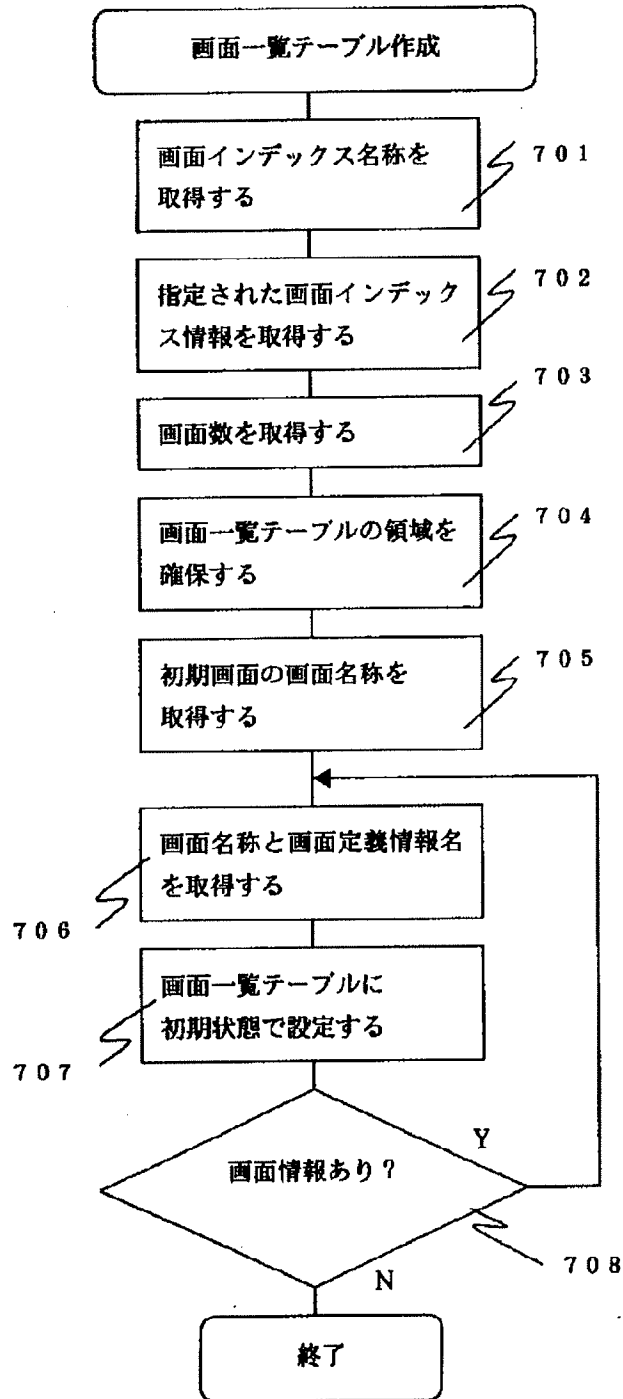
【図10】

図 10

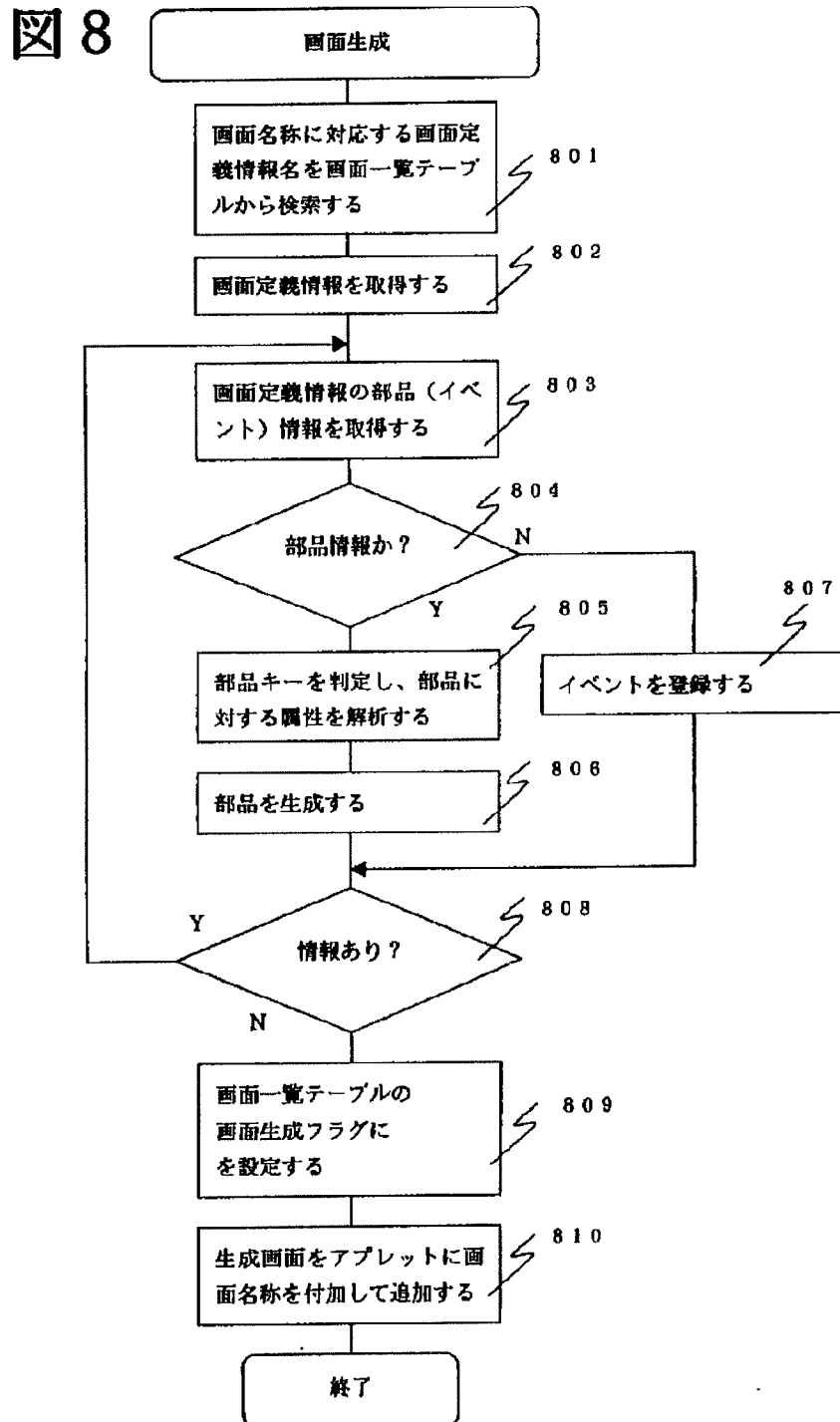


【図7】

図7



【図8】



【図 9】

図 9

